Accurate angular velocity estimation with an event camera

Gallego, Guillermo ; Scaramuzza, Davide

Abstract: We present an algorithm to estimate the rotational motion of an event camera. In contrast to traditional cameras, which produce images at a fixed rate, event cameras have independent pixels that respond asynchronously to brightness changes, with microsecond resolution. Our method leverages the type of information conveyed by these novel sensors (i.e., edges) to directly estimate the angular velocity of the camera, without requiring optical flow or image intensity estimation. The core of the method is a contrast maximization design. The method performs favorably against ground truth data and gyroscopic measurements from an Inertial Measurement Unit, even in the presence of very high-speed motions (close to 1000 deg/s).

DOI: https://doi.org/10.1109/lra.2016.2647639

Accurate Angular Velocity Estimation With an Event Camera

Guillermo Gallego and Davide Scaramuzza

***Abstract*—We present an algorithm to estimate the rotational motion of an event camera. In contrast to traditional cameras, which produce images at a fixed rate, event cameras have independent pixels that respond asynchronously to brightness changes, with microsecond resolution. Our method leverages the type of information conveyed by these novel sensors (i.e., edges) to directly estimate the angular velocity of the camera, without requiring optical flow or image intensity estimation. The core of the method is a contrast maximization design. The method performs favorably against ground truth data and gyroscopic measurements from an Inertial Measurement Unit, even in the presence of very high-speed motions (close to 1000 deg/s).**

***Index Terms*—Computer Vision for Other Robotic Applications, Localization**

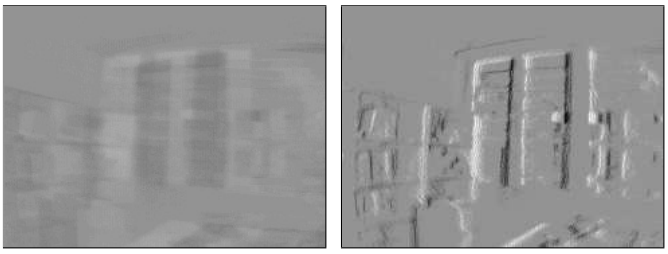
Supplementary Material

A video showing the performance of our method on several sequences is available at: http://youtu.be/v1sXWoOAs_0

I. Introduction

EVENT cameras [1] are biologically inspired sensors that overcome some of the limitations of traditional cameras: they have a very fast response (in the order of microseconds), very high dynamic range and require low power and bandwidth. These advantages come from their fundamentally different principle of operation: they have independent pixels that sense and asynchronously transmit brightness changes (called "events"). Hence, their output is not a sequence of frames at fixed rate but rather a spatially sparse, asynchronous stream of events. Event cameras offer great potential for high-speed robotics and applications with large illumination variations. However, new methods have to be designed to cope with their unconventional output.

In this paper we are interested in unlocking the high-speed capabilities of the sensor to estimate ego-motion. In particular, we focus on the restricted but important case of 3D rotational motions (i.e., estimating the angular velocity of the camera). Orientation estimation, besides being an important topic on its own, is a recurrent topic in visual odometry scenarios, where the camera might move with negligible translation with respect to the depth of the scene, potentially causing a breakdown of the system if the 3D map used for localization falls out of the field of view of the camera. Orientation estimation also finds applications in camera stabilization [2] and in panoramic image creation [3].



(a) Image of accumulated events without motion estimation.

(b) Image of accumulated events, rotated according to motion.

Fig. 1: Rotational motion estimation by contrast maximization. Events accumulated in a small time interval (e.g., ms) taking into account the rotational motion of the event camera produce images with stronger edges (Fig. 1b), i.e., larger contrast, than those that do not take into account the motion or incorrectly estimate it (Fig. 1a).

Contribution: This paper presents a novel method to estimate 3D rotational motion of an event camera. The method aligns events corresponding to the same scene edge by maximizing the strength of edges obtained by aggregating motion-warped events. Our method works directly on the event stream, hence it does not require the estimation of intermediate quantities such as image intensity or optical flow like other approaches. Besides the notable accuracy and robustness of the proposed method, its most interesting insight is that it admits an intuitive formulation in terms of contrast maximization (Fig. 1), and that contrast is a basic signal statistic with broad applicability. Thus, the method carries a new design philosophy for event-based algorithms.

II. Related Work

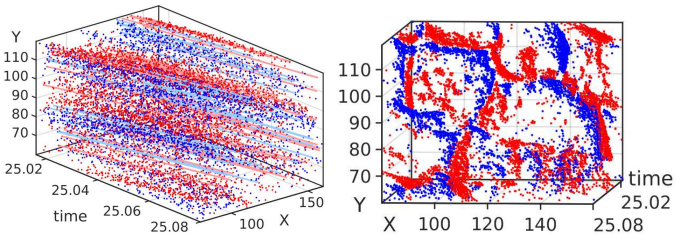
There are few works on 3D orientation estimation with event cameras. This may be due to the following facts: research is dominated by standard (frame-based) cameras, event cameras have been commercially available only recently [1] and they are still expensive sensors since they are at an early stage of development.

A generic message passing algorithm within an interacting network to jointly estimate several quantities (called "maps"), such as, rotational ego-motion, image intensity and optical flow from a stream of events was proposed by Cook et al. [4].

Manuscript received: September, 10, 2016; revised December, 6, 2016; accepted December, 27, 2016. This paper was recommended for publication by Associate Editor E. M. Mouaddib and Editor F. Chaumette upon evaluation of the reviewers' comments. This work was supported in part by the DARPA FLA Program, in part by the National Centre of Competence in Research Robotics through the Swiss National Science Foundation, in part by the SNSF-ERC Starting Grant and the UZH Forschungskredit.

The authors are with the Robotics and Perception Group, University of Zurich, Zurich 8050, Switzerland—http://rpg.ifi.uzh.ch.

Digital Object Identifier (DOI): 10.1109/LRA.2016.2647639



(a) Events (dots) and the trajectories that they follow. (b) Events visualized along the trajectories in Fig. 2a.

Fig. 2: Visualization of the events (positive (blue dots) and negative (red dots)) in the image plane vs. time (50 ms).

(a) Blurred image. The blur kernel (PSF) is shown in a corner. (b) Restored image (blur corrected, but still with artifacts).

Fig. 3: In standard cameras, motion-compensated images (right) have higher contrast than uncompensated ones (left). A similar idea applies to event images (Fig. 1).

The algorithm is not a traditional, feed-forward pipeline but can be interpreted as a joint estimation of optical flow and image intensity from the event stream while, at the same time, enforcing that the resulting quantities (e.g., optical flow field) are consistent with a global constraint: the estimated motion must be rotational.

More recently, Kim et al. [3] presented a parallel tracking-and-mapping filter-based system that estimated the 3D orientation of an event camera while generating high-resolution panoramas of natural scenes. The tracking thread estimated rotational motion by means of a particle filter using the event stream and a given intensity image (the panorama).

Conradt [5] presented a simple algorithm to extract optical flow information from event cameras, and as an application, he showed that it can be used for ego-motion estimation. He first computed the optical flow and then calculated the (3D) angular velocity that best explained such a 2D flow.

All previous works require an auxiliary variable such as optical flow or image intensity to estimate angular velocity. For example, [4] and [5] estimate angular velocity given or together with optical flow, whereas [3] requires an intensity image to provide a likelihood for the events undergoing a candidate rotational motion. In contrast, our method shows that angular velocity can be estimated directly, without having to reconstruct image intensity or optical flow.

III. Methodology

A. Intuitive Explanation of the Approach

Event cameras have independent pixels that respond asynchronously to brightness changes, which are due to moving edges, i.e., intensity gradients, in the image plane. Thus, these cameras output a sequence of asynchronous "events" (Fig. 2a). Each event is described by a tuple $e_k = (x_k, y_k, t_k, \pm_k)$, where $\mathbf{x}_k = (x_k, y_k)^\top$ and t_k are the spatio-temporal coordinates of the brightness change and $\pm_k$, the binary event polarity, is the sign of the brightness change (the color of the dots in Fig. 2a). Events are time-stamped with microsecond resolution (t_k).

Fig. 2a shows the output of a rotating event camera over a small time interval. Looking at the events only (i.e., omitting the overlayed trajectories) it seems that the information of the moving edges that triggered the events is unintelligible. In the example, the edges moved approximately along linear trajectories in the space-time volume of the image plane (Fig. 2a), and it is only when the events are observed along such trajectories that the edge structure is revealed (Fig. 2b). Moreover, Fig. 2 provides a key observation: the events along a trajectory are triggered by the same scene edge (they are corresponding events) and they all have the same polarity[1] Thus, we can use the event polarities along trajectories to analyze the edge structure, and therefore, reveal the unknown camera motion. In particular, we just consider the sum of the polarities along each trajectory, with as many trajectories as pixels in the image plane. If we naively sum the event polarities pixelwise (along trajectories parallel to the time axis), we will generate an event "image" showing the trace of the edges in the scene as they moved through the image plane (Fig. 1a). Observe that this is analogous to the motion blur effect in standard cameras, caused by large exposure times (Fig. 3a). The shapes of such traces provide visual cues of the motion that caused them, and once such a motion has been estimated, usually represented by a Point Spread Function (PSF), as shown in the bottom-left of Fig. 3a, a sharp image can be obtained from the blurred one by compensating for the motion, a process known as deconvolution (Fig. 3b). Similarly, if we are able to estimate the motion of an event camera, e.g., by searching for the trajectories that satisfy the above-mentioned property of corresponding events, we may compensate for it. The resulting event image, obtained by summing the event polarities along the pixel trajectories induced by the true camera motion, does not suffer from accumulation blur (Fig. 1b), and consequently, has stronger edges than those of the uncompensated one (Fig. 1a). Hence, a strategy to estimate the ego-motion of the camera is to search for the motion and scene parameters that maximize the strength of the motion-compensated edges. In the case of rotational motions the problem simplifies since no scene parameters such as depth are needed to represent the trajectories in the image plane; the problem solely depends on the motion parameters (the angular velocity of the camera).

Here we present a method that exploits the previous ideas to estimate the motion undergoing a rotating event camera, namely by measuring the edge strength using image contrast, and therefore, our method can be interpreted as motion estimation by contrast maximization. Fig. 4 summarizes our approach, which we describe in the next sections. First, we show how to create an event image (Section III-B), how

[1] This holds except when the motion changes direction abruptly, which can be detected since the camera triggers no events while it is at rest between both states.

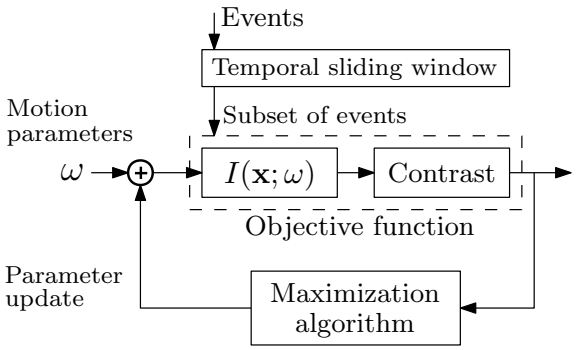


Fig. 4: Block diagram of the method: events from the camera are rotated according to a candidate angular velocity $\boldsymbol{\omega}$, which is iteratively refined by a maximization algorithm on the contrast of the images of rotated events $I(\mathbf{x};\boldsymbol{\omega})$.

to displace events according to 3D rotational motion (Section III-C) and how to measure the strength of the (motion-compensated) edges (Section III-D). Then, we discuss the maximization strategy and how to process an entire event stream (Section III-E).

B. From Events to Event Images

Event images, such as those in Fig. 1, are formed by adding event polarities along candidate trajectories in the image plane (Fig. 2a). More specifically, given a set of events $\mathcal{E} = \{e_k\}_{k=0}^{N-1}$ triggered in a small time interval $[0, \Delta t]$, the event image formed by polarity addition along trajectories parallel to the time axis is given by

$$I(\mathbf{x}) = \sum_{k=0}^{N-1} \pm_k \, \delta(\mathbf{x} - \mathbf{x}_k), \tag{1}$$

where, to later allow for arbitrary (sub-pixel) trajectories, we represent images as functions $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, and δ is the Dirac delta. Thus, the intensity I at pixel $\mathbf{x}$ is the sum of the polarities of the events that fired at the pixel location $\mathbf{x} = \mathbf{x}_k$.

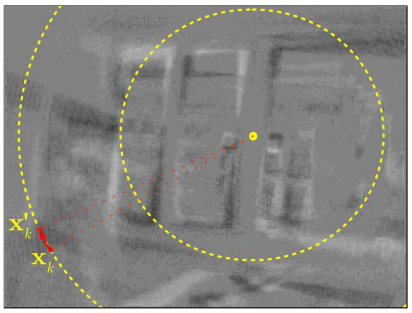
Event images corresponding to arbitrary pixel trajectories are formed by displacing the events $\mathbf{x}_k \mapsto \mathbf{x}'_k$ before their polarities are added using (1), i.e., the trajectories are mapped to lines parallel to the time axis before polarity addition.

C. Motion Compensation of Events

Under a rotational camera motion, the trajectories of points in the image plane are parametrized by the motion parameters. Thus, given an angular velocity $\boldsymbol{\omega}(t) \in \mathbb{R}^3$, a point $\mathbf{x}_0$ in the image plane will describe a path $\mathbf{x}(t) = \mathbf{W}(\mathbf{x}_0, \boldsymbol{\omega}(t))$, which is represented by a warp $\mathbf{W}$. In calibrated coordinates (intrinsic parameters and lens distortion removed), such a warp is described by a 2D homography, defined in terms of the matrix of the 3D rotational motion $\mathtt{R}(t)$ [6, p.204]:

$$\mathbf{x}(t) \sim \mathtt{R}(t)\mathbf{x}_0,$$

where $\sim$ means equality up to a non-zero scale factor typical of homogeneous coordinate representation. The rotation $\mathtt{R}(t)$ is obtained from the angular velocity and the motion duration using the matrix exponential. More specifically, consider, without loss of generality, that $t \in [0, \Delta t]$, a small time interval



(a) Event warp overlaid on the event image $I(\mathbf{x})$, before motion compensation. $I(\mathbf{x})$ is obtained by aggregating event polarities in time, as specified by (1). Blur increases with the distance to the center of rotation, which is marked with a yellow disk.

(b) Image of rotated events, i.e., after motion compensation. $I(\mathbf{x};\boldsymbol{\omega})$ is given by (4) with the optimal $\boldsymbol{\omega}$. Observe how sharp the edges are everywhere in the image, even far away from the center of rotation.

Fig. 5: Warp $\mathbf{W}$ in (2)-(3) mapping a point $\mathbf{x}_k$ to its rotated position $\mathbf{x}'_k$. The example corresponds to a rotation approximately around the optical axis (Z camera axis).

so that the angular velocity is constant, and that $\mathtt{R}(0)$ is the identity. Then, the rotation $\mathtt{R}(t)$ is given by [7, p.26]:

$$\mathtt{R}(t) = \exp\left(\widehat{\boldsymbol{\omega}} t\right),$$

where $\widehat{\mathbf{a}}$ is the cross-product matrix, i.e., the 3×3 skew-symmetric matrix such that $\widehat{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$, $\forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^3$. The warp is, in homogeneous calibrated coordinates, given by

$$\mathbf{W}(\mathbf{x}_0; \boldsymbol{\omega}, t) \sim \exp\left(\widehat{\boldsymbol{\omega}} t\right) \mathbf{x}_0, \tag{2}$$

where we explicitly noted the three elements involved: the point to be warped $\mathbf{x}_0$, the angular velocity $\boldsymbol{\omega}$ and the duration of the motion t. Observe that $\mathbf{W}(\mathbf{x}_0; \boldsymbol{\omega}, 0) = \mathbf{x}_0$ is the identity warp, that is, $\mathbf{x}(0) = \mathbf{x}_0$.

We use the above-defined warp to rotate events in the image plane: given an angular velocity $\boldsymbol{\omega}$, an event at $\mathbf{x}_k$ is mapped to the point

$$\mathbf{x}'_k = \mathbf{W}(\mathbf{x}_k; \boldsymbol{\omega}, t_k - t_0), \tag{3}$$

where t_k is the time of the event and t_0 is the time of the first event in the subset $\mathcal{E}$, which is used as reference. Observe that the 3D rotation angle of each event is different, $\boldsymbol{\theta}_k = (t_k - t_0)\boldsymbol{\omega}$, since it depends on the event time t_k; otherwise, if the rotation angle was the same for all events, it would not be possible to compensate for the motion.

By rotating all events in the set $\mathcal{E}$ and adding their polarities, an image of (un-)rotated events is obtained:

$$I(\mathbf{x}; \boldsymbol{\omega}) = \sum_k \pm_k \, \delta(\mathbf{x} - \mathbf{x}'_k(\boldsymbol{\omega})). \tag{4}$$

In practice, the Dirac delta is replaced by an approximation such as a Gaussian to allow us to add the polarities of sub-pixel rotated events $\mathbf{x}'_k(\boldsymbol{\omega})$ to produce the value at a pixel location, $I(\mathbf{x})$ (see details in Section IV).

D. Measuring the Edge Strength of an Event Image

The goal of our ego-motion method is to use $I(\mathbf{x};\boldsymbol{\omega})$ to estimate the $\boldsymbol{\omega}$ that aligns all corresponding events (those that were triggered by the same scene point) to the same (un-rotated) image point, thus effectively removing the accumulation blur.

Given a subset of events, we cast the ego-motion estimation problem into an optimization one: obtain the angular velocity $\boldsymbol{\omega}$ that optimizes some distinctive characteristic of motion-compensated event images. But what are such distinctive characteristics? Drawing an analogy with standard cameras, motion-compensated intensity images look *sharper* and have *higher contrast* than the blurred (uncompensated) ones, as shown in Fig. 3b. This is intuitive, since blur is given by the convolution of the original (sharp) image with a low-pass filter, and restoration consists in inverting such an operation, that is, high-pass filtering. Sharpening is nothing but increasing the contrast along the edges, making the light side of the edge lighter and the dark side of the edge darker.

In the same way, distinctive characteristics of motion-compensated event images are that they look sharper and have higher contrast than uncompensated ones (cf. Figs. 5a and 5b). In fact, both, sharpness and contrast, are related. Hence, we will use contrast to measure the quality of an event image. In general, contrast quantifies the amount by which the oscillation (or difference) of a signal stands out from the average value (or background). Several contrast metrics are available (see [8]). The Weber contrast is defined locally, as $C_W \doteq (I - I_b)/I_b$, with I and I_b representing the uniform intensities of a small image object and its large adjacent background, respectively. The Michelson contrast [9], defined as $C_M \doteq (I_{\max} - I_{\min})/(I_{\max} + I_{\min})$ with $I_{\max}$ and $I_{\min}$ representing the highest and lowest intensity, is suitable for images with periodic patterns where there is no large area of uniform intensity. We measure the contrast of an image by means of its variance, as defined in Appendix A,

$$\operatorname{Var}\left(I(\boldsymbol{\omega},\mathcal{E})\right) \doteq \frac{1}{|\Omega|}\int_{\Omega}\left(I(\boldsymbol{\omega},\mathcal{E})(\mathbf{x}) - \mu(I(\boldsymbol{\omega},\mathcal{E}))\right)^2 d\mathbf{x}, \quad (5)$$

which is a measure of the spread or concentration of the image values around the mean intensity and it does not depend on the spatial distribution of contrast in the image. Alternative contrast metrics such as the RMS (Appendix A) or different p-norms, $C_p \propto \int_{\Omega} |I(\boldsymbol{\omega},\mathcal{E})(\mathbf{x}) - \mu(I(\boldsymbol{\omega},\mathcal{E}))|^p d\mathbf{x}$, with $p \geq 1$, are also possible. We opt for the variance (2-norm in (5)) since it performs better than other metrics.

Since event images add the polarity of the events, which are caused by scene edges, the contrast of the event image measures the strength of the edges. Corresponding events (Section III-A) have the same polarity, so a candidate $\boldsymbol{\omega}$ that aligns corresponding events (i.e., compensates for motion) will sum their polarities, producing stronger edges, and therefore, increasing the contrast.

E. Ego-Motion Estimation by Contrast Maximization

The contrast (5) of the image of rotated events $I(\boldsymbol{\omega},\mathcal{E})$ (4) provides a measure of the goodness of fit between the event data $\mathcal{E}$ and a candidate angular velocity $\boldsymbol{\omega}$. Hence, we can use it in the above-mentioned optimization framework (Fig. 4): by maximizing the contrast (i.e., quality) of the image of rotated events we will estimate the motion parameters that best compensate for the rotational motion of the camera, i.e., those that best describe the ego-motion.

The contrast (5) is a non-linear function of the unknown variable $\boldsymbol{\omega}$. It is unlikely that a closed-form solution to the contrast maximization problem

$$\max_{\boldsymbol{\omega}} \operatorname{Var}\left(I(\boldsymbol{\omega},\mathcal{E})\right)$$

exists. Therefore, we use standard iterative non-linear algorithms to optimize the contrast. In particular, we use the non-linear conjugate gradient (CG) method by Fletcher and Reeves [10], CG-FR.

To process an entire stream of events, we use a temporal observation window consisting of a subset of events $\mathcal{E}_m$. We process the subset (i.e., maximize contrast) and then shift the window, thus selecting more recent events. The angular velocity estimated using $\mathcal{E}_m$ provides an initial guess for the angular velocity of the next subset, $\mathcal{E}_{m+1}$, thus effectively assuming a constant velocity motion model. This scheme works very well in practice (in spite of the local convergence properties of standard optimization algorithms) since each subset $\mathcal{E}_m$ usually spans a very small time interval, and therefore, the angular velocity does not significantly change between consecutive event subsets.

The answer to the question of how to choose the number of events in each subset $\mathcal{E}_m$ and how to shift the window is application-dependent: the two principal strategies consist of using a fixed time interval Δt and shift $\Delta t'$ or using a fixed number of events per subset N and per shift N'. The first one might be the choice of applications that must provide angular velocity estimates at a fixed rate. Since event cameras are data-driven sensors, whose output depends on the amount of apparent motion, we opt for the second strategy (fixed number of events) because it preserves the data-driven nature of event cameras: the rate of ego-motion estimates will be proportional to the event rate, that is, to the apparent motion of the scene.

IV. Algorithm Details

This section describes details of the proposed method. The reader not interested in the details can jump to Section V.

An efficient implementation of the method requires providing to the optimization algorithm not only the contrast but also its gradient with respect to $\boldsymbol{\omega}$. For completeness, such formulas are given in Appendix B.

Formula (4) is an idealized description of the image of rotated events. In practice, a digital image is synthesized, so the image domain Ω has to be discretized into pixels and the two-dimensional Dirac delta has to be replaced by a suitable approximation, as in forward mapping of spatial transformations [11, ch.3]. The simplest one consists in a single-pixel update: the rotated event at point $\mathbf{x}'_k(\boldsymbol{\omega})$ only

updates the value of the accumulated polarity at the nearest pixel. However, this is a crude approximation that produces undesirable rounding effects ("aliasing", in the terminology of line rasterization in Computer Graphics). Instead, we use bilinear voting, where the polarity of the rotated event $\mathbf{x}'_k(\boldsymbol{\omega})$ is used to update the accumulated polarities $I(\mathbf{x};\boldsymbol{\omega})$ of the four nearest pixel locations; with update weights that take into account the distances from $\mathbf{x}'_k(\boldsymbol{\omega})$ to the integer pixel locations $\mathbf{x}$, similarly to bilinear interpolation.

To improve robustness against noise, we smooth the synthesized image of rotated events using a Gaussian filter with a small standard deviation ($\sigma = 1$ pixel). That is, we maximize the contrast of $I_\sigma(\mathbf{x};\boldsymbol{\omega}) = I(\mathbf{x};\boldsymbol{\omega}) * G_\sigma(\mathbf{x})$. This diffusion operation spreads the polarity of the rotated event $\mathbf{x}'_k(\boldsymbol{\omega})$ beyond its four neighbors, since the convolution replaces the δ in (4) by a broader kernel G_σ. A smoother event image yields a smoother contrast function, which is in turn easier to optimize (faster convergence) since optimization algorithms exploit local smoothness.

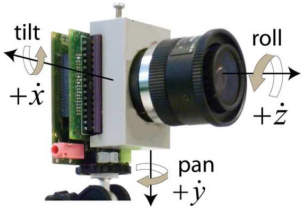
To speed up the algorithm, the rotation matrix in the warp (2) is replaced by its first order approximation, $\mathtt{R}(t) = \exp(\widehat{\boldsymbol{\omega}}t) \approx \mathtt{Id} + \widehat{\boldsymbol{\omega}}t$, where $\mathtt{Id}$ is the identity matrix. This is a reasonable approximation since the incremental rotation between consecutive event images is small due to the very high temporal resolution (microseconds) of event cameras and the small number of events in the subset $\mathcal{E}_m$ (typically in the order of $10\,000$ events); hence the subset of events $\mathcal{E}_m$ spans a very small time interval Δt, which multiplied by the angular velocity gives a very small angle. This approximation yields simplified formulas for the rotation of an event: the warp becomes as simple as a sum and a cross product $\mathbf{W}(\mathbf{x};\boldsymbol{\omega},t) \approx \mathbf{x} + t\boldsymbol{\omega} \times \mathbf{x}$ (in homogeneous coordinates).

The mean of the image of rotated events is constant: $\mu(I(\mathbf{x};\boldsymbol{\omega})) = (\sum_{k=0}^{N-1} \pm_k)/N_p$, and, as the formula shows, it does not depend on $\boldsymbol{\omega}$; it only depends on the balance of polarities in the subset of events $\mathcal{E}_m$ used to generate the image, divided by the number of pixels N_p. In generic scenes, because both dark-to-bright and bright-to-dark edges are, typically, of the same magnitude and equally probable, the corresponding events are both positive and negative (see Figs. 1b, 5b) with equal probability. Hence the balance $\sum_{k=0}^{N-1} \pm_k \approx 0$, and so is the mean, $\mu(I(\mathbf{x};\boldsymbol{\omega})) \approx 0$. The fact that the mean of $I(\mathbf{x};\boldsymbol{\omega})$ is approximately zero may be used, if desired, to simplify the contrast function, replacing the variance of the image by the mean square value.

We use the standard optimization methods in the scientific library GNU-GSL to implement the contrast maximization. The CG-FR algorithm converges in, typically, 2 to 4 line searches. Other methods, such as CG-Polak-Riviere and the quasi-Newton method BFGS, give similar results.

V. Experiments

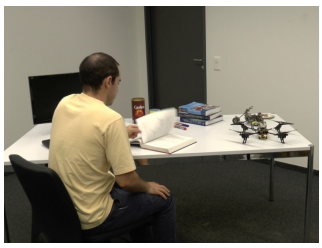
In this section, we assess the accuracy of our orientation ego-motion estimation method both quantitatively and qualitatively on different challenging sequences. The results show that our method produces reliable and accurate angular velocity estimates.



(a) The DAVIS240C sensor and rate gyro axes definitions. Image adapted from [2].

(b) `poster`. Images (b)-(d) are courtesy of [13].

(c) `boxes`



(d) `dynamic`

Fig. 6: DAVIS sensor and scenes of the evaluation sequences.

The event camera used to acquire the datasets was the DAVIS [12], which has a spatial resolution of 240×180 pixels, a temporal resolution of microseconds and a very high dynamic range (130 dB). The DAVIS combines in the same pixel array an event sensor and a standard, frame-based sensor. However, our algorithm uses only the event stream and not the frames. The DAVIS also has an integrated Inertial Measurement Unit (IMU). The rate gyro axes definitions of the DAVIS is illustrated in Fig. 6a, with the IMU axes aligned with the camera axes. The angular rates around the X, Y and Z axes of the camera are called tilt (up/down), pan (left/right) and roll (optical axis rotation), respectively.

A. Accuracy Evaluation

To assess the accuracy and robustness of our method we evaluated it on three different sequences from [13]: `poster`, `boxes` and `dynamic` (see Fig. 6). The `poster` scene features a textured wall poster; the `boxes` scene features some boxes on a carpet, and the `dynamic` scene consists of a desk with objects and a person moving them. All sequences contain, in addition to the event stream, angular velocity measurements that we use for comparison: gyroscope data from the IMU of the DAVIS and ground truth pose measurements from a motion capture system (mocap), from which angular rates are obtained. The IMU operates at 1 kHz, and the motion capture system at 200 Hz. The sequences were recorded hand-held. Each sequence has a 1 minute length and contains about 100-200 million events. Each sequence starts with rotations around each camera axis, and then is followed by rotations in all 3-DOFs. Additionally, the speed of the motion increases as the sequence progresses.

Fig. 7 shows the comparison of the results of our method against ground truth on the `poster` sequence. Fig. 7-middle shows the results on the entire sequence. Observe the increasing speed of the motion, with excitations close to ± 1000 deg/s. The results provided by our method are very accurate,

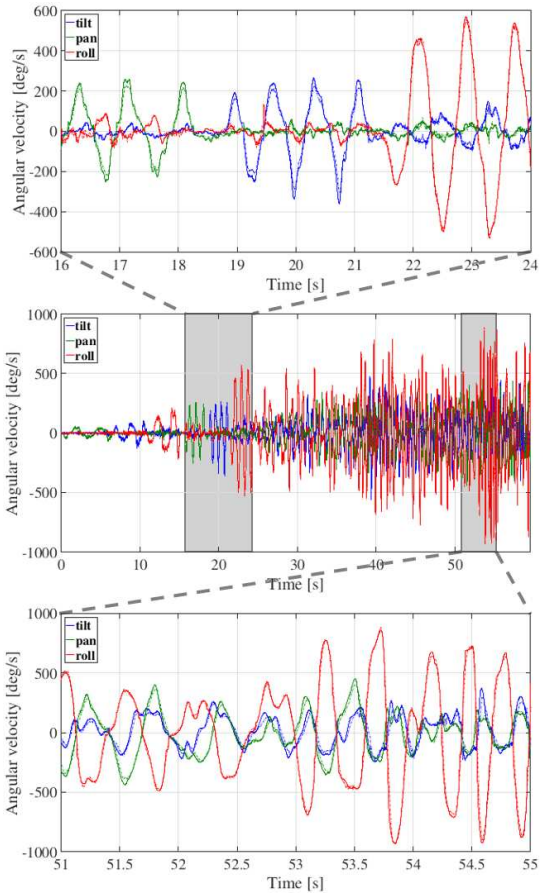


Fig. 7: `poster` sequence. Comparison of the estimated angular velocity (solid line) against ground truth from the motion capture system (dashed line). Whole sequence (middle) and zoomed-in plots of shaded regions (top and bottom).

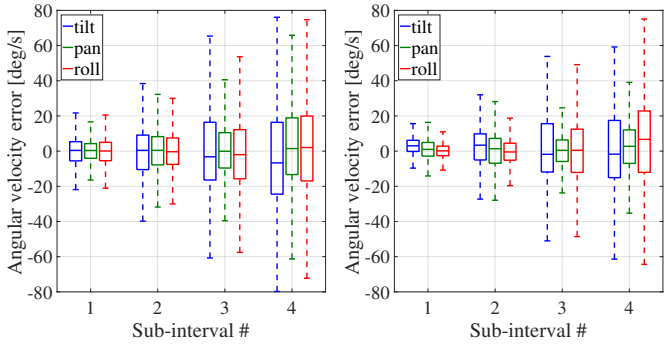


Fig. 8: `poster` sequence. Error box plots, by intervals of 15s. Left: Estimated vs. mocap. Right: Estimated vs IMU.

as highlighted by the very small errors: the lines of our method and those of the ground truth are almost indistinguishable at this scale. To better appreciate the magnitude of the error, we zoomed-in at the shaded regions of Fig. 7-middle in Figs. 7-top and bottom. Fig. 7-top shows a segment of 8 seconds duration, with rotations dominantly around each axis of the event camera: first pan, then tilt, and, finally, roll. Fig. 7-bottom shows a $4\,\mathrm{s}$ segment at the end of the sequence with the largest combined rotations in all 3 axes.

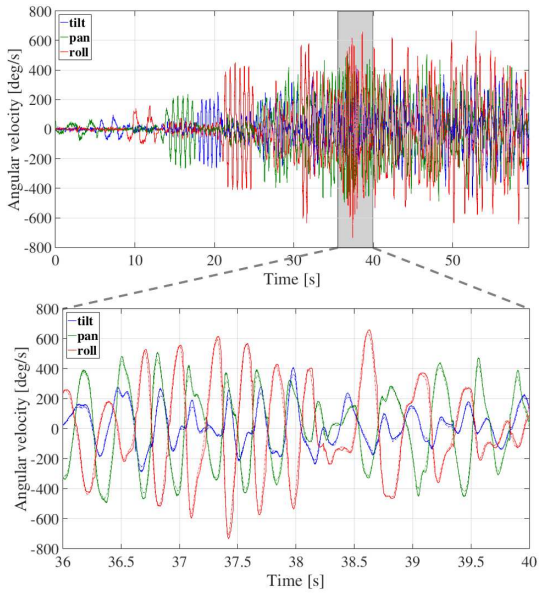
The plots for the same `poster` sequence comparing our method against the IMU are very similar, and, therefore, are not included for brevity. Instead, we provide box plots with statistics about the errors of our method against both, mocap and IMU, in Fig. 8. Since the sequence presents increasing motion, we split the analysis of the statistics in intervals, each of which lasts 15 seconds. Recall that the bottom and top of a box are the first and third quartiles, and the line inside the box is the second quartile (i.e., the median). The markers outside the box are the minimum and maximum. We observe the increasing trend of the motion speed also in the error box plots, suggesting that there is a relative dependency: the larger the motion, the larger the error can be. However, note that the errors are small relative to the "size" of the motion: we report standard and maximum deviations of approximately $20\,\mathrm{deg/s}$ and $80\,\mathrm{deg/s}$, respectively, with respect to peak excursions close to $1000\,\mathrm{deg/s}$, which translate into $2\,\%$ and $8\,\%$, respectively.

Figs. 9 and 10 summarize our results for the `boxes` and `dynamic` sequences, respectively. For the `boxes` sequence, Fig. 9a-top shows the comparison of our method against ground truth over the whole sequence. The estimated motion with our method is identical to ground truth at this scale. Fig. 9a-bottom provides a zoom into the comparison plot, during a 4 second segment with high-speed motion (angular speeds of more than $\pm 600\ \mathrm{deg/s}$). Even at this zoom level, the lines of both our method and ground truth are almost identical. A better visualization of the magnitude of the errors is provided in the box plots of Fig. 9b. This figure also shows the comparison of our method against the IMU, and it is analogous to Fig. 8. As it can be observed, our method compares favorably in both cases: ground truth and IMU.

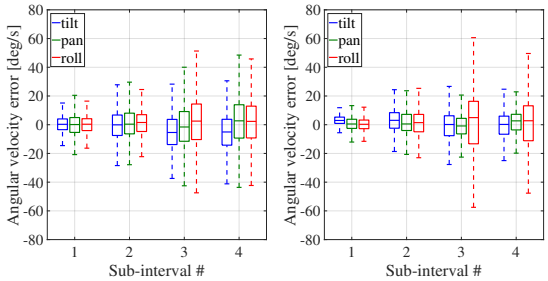
Figs. 10a-top and 10a-bottom compare our method against ground truth over the entire `dynamic` sequence and over a six-second segment (zoomed-in view) featuring high-speed motions of up to $500\,\mathrm{deg/s}$, respectively. This sequence depicts a desk scene, with events being generated by both the static objects and a moving person. The events caused by the moving person do not fit the rotational motion model of a static scene. However, the motion of the person is slow compared to the temporal resolution of the event cameras, and, most of the time, our algorithm is not affected by such motion. When a significant amount of the events (typically, $20\,\%$ or more) are triggered by the moving person, as shown in half of the image plane in Fig. 11-left, the performance of the ego-motion algorithm is affected since no outlier rejection has been implemented. In Fig. 11-right, the estimated angular velocity (pan and tilt) deviates from ground truth at $t \approx 32\,\mathrm{s}$. Nevertheless, the box plots in 10b show that the errors of our method against ground truth and against the IMU remain small for this sequence.

B. Computational Performance

Next, we provide some order of magnitude of the computational effort required by the method. The algorithm was implemented in C++, without paying attention at optimizing the code for real-time performance. The core of the algorithm is the computation of the image of rotated events $I(\mathbf{x};\boldsymbol{\omega})$ in (4) and its derivatives (Eq. (9) in Appendix B). For a subset of $15\,000$ events, this takes $2.7\,\mathrm{ms}$ on a standard laptop

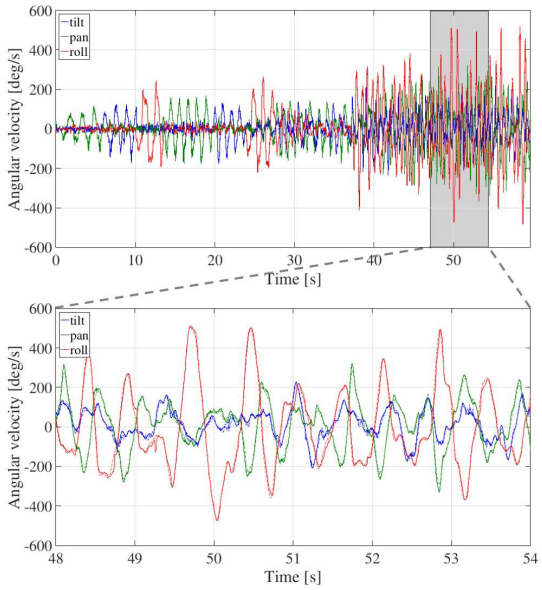


(a) Whole sequence (top) and zoom-in of shaded region (bottom).

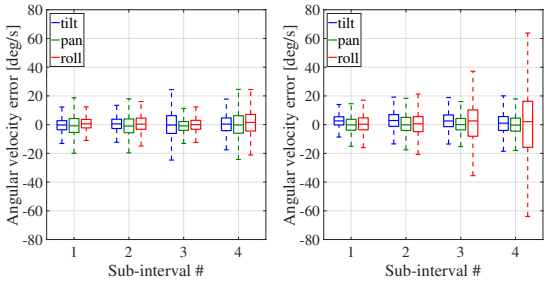


(b) Error box plots, by intervals of 15s. Left: Estimated vs mocap. Right: Estimated vs IMU.

Fig. 9: `boxes` sequence. Comparison of the estimated angular velocity (solid line) against ground truth from the motion capture system (mocap) (dashed line).



(a) Whole sequence (top) and zoom-in of shaded region (bottom).



(b) Error box plots, by intervals of 15s. Left: Estimated vs mocap. Right: Estimated vs IMU.

Fig. 10: `dynamic` sequence. Comparison of the estimated angular velocity (solid line) against ground truth from the motion capture system (mocap) (dashed line).

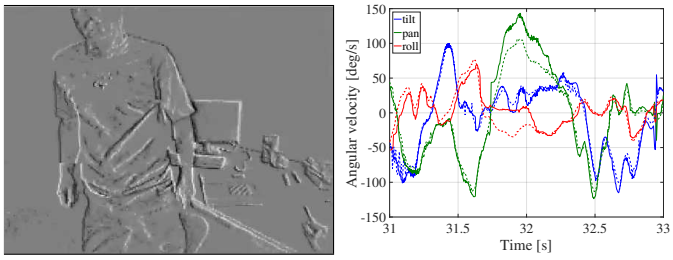


Fig. 11: `dynamic` sequence. Detail of person moving close to the camera. Left: image of rotated events. Right: estimated angular velocity (solid line) vs. ground truth (dashed line).

with an Intel(R) Core(TM) i7-3720QM CPU @ 2.60GHz running single-threaded. This operation is carried out multiple times within the optimization algorithm. The method may be accelerated for real-time performance.

Since the contrast (5) is computed using the image of rotated events (4), the computational complexity of the method depends on both the number of rotated events and the event image resolution. The method scales linearly with respect to both. Hence, to reduce the computational cost, one may (i) compute the contrast using fewer pixels, (ii) rotate fewer events or (iii) apply both strategies. An appealing option is to maximize the contrast of some regions of the image plane. It suffices to select the most informative regions (those with the largest density of events), as in direct methods for visual odometry [14] and SLAM [15], i.e., it suffices to select the events that most likely will be rotated to the regions of the image that will present the highest contrast, which would then be tracked in time, for efficiency.

C. Discussion

During the experiments, we noticed that, in general, roll estimation is more subjective to errors than the estimation of pan and tilt motions because the apparent motion of the center of the image is smaller than the periphery, and, therefore, there is a lack of events close to the center of rotation, with events mostly appearing far away, in the periphery of the image.

In all sequences, the event camera was moved in front of the scene, about $1.5\,\mathrm{m}$ away or more. The motions were hand-held and are inevitably corrupted by translation. However, the translational component of the motion was negligible with respect to the mean scene depth (e.g., distance to the desk), that is, motion was dominantly rotational, satisfying the hypothesis of our framework. We tested the algorithm on sequences with significant translation and, as expected for any

algorithm designed for rotation-only motions, the algorithm provided an incorrect motion estimate since it tried to explain the translation using a rotation.

VI. Conclusions

We have developed a method that estimates the rotational motion of an event camera. We have tested the algorithm on sequences with millions of events, and the results obtained are very accurate, with angular velocity errors of $2\,\%$ (standard deviation) and $8\,\%$ (maximum). Due to the high temporal resolution of the sensor, our method is able to track very high-speed motions ($\approx 1000\,\mathrm{deg/s}$). Additionally, due to the sliding window approach, our method can provide angular velocity estimates at the rate implied by the event resolution ($1\,\mu\mathrm{s}$; sliding the window by one event). Besides the remarkable accuracy and robustness of the method, we believe that its most interesting characteristic is its design philosophy: motion estimation by means of edge alignment in terms of contrast maximization.

Appendix A
Moments of an Image (Continuous Framework)

The mean and variance of an image $F : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ are

$$\mu_F \doteq \mu(F) = \frac{1}{|\Omega|}\int_\Omega F(\mathbf{x})d\mathbf{x}, \tag{6}$$

$$\sigma_F^2 \doteq \mathrm{Var}\,(F) = \frac{1}{|\Omega|}\int_\Omega (F(\mathbf{x})-\mu_F)^2 d\mathbf{x}, \tag{7}$$

respectively, where $|\Omega|$ is the area of the image domain. These formulas can be obtained by interpreting the values $\{F(\mathbf{x})\}_{\mathbf{x}\in\Omega}$ as providing infinitely many samples of a random variable and using the moments of such a random variable to define the moments of the image. Hence, as it agrees with the intuition, the mean μ_F is the average value of the image F over the domain Ω, and the variance is the average spread (i.e., dispersion) of F around μ_F, over the domain Ω. For more details, see [16].

The mean square of an image is defined in the usual way, in terms of the mean and the variance: $\mathrm{MS} = \mathrm{RMS}^2 = \mu_F^2 + \sigma_F^2$, that is, $\mathrm{MS} = \frac{1}{|\Omega|}\int_\Omega F^2(\mathbf{x})d\mathbf{x}$.

Appendix B
Derivative of the Contrast Metric

Efficient optimization schemes make use of the derivative of the objective function to search for ascent/descent directions. For the proposed contrast metric (5), the derivative is

$$\frac{\partial}{\partial\boldsymbol{\omega}}\mathrm{Var}\,(I(\mathbf{x};\boldsymbol{\omega})) \overset{(7)}{=} \frac{1}{|\Omega|}\int_\Omega 2\rho(\mathbf{x};\boldsymbol{\omega})\frac{\partial\rho(\mathbf{x};\boldsymbol{\omega})}{\partial\boldsymbol{\omega}}d\mathbf{x}, \tag{8}$$

where $\rho(\mathbf{x};\boldsymbol{\omega}) \doteq I(\mathbf{x};\boldsymbol{\omega}) - \mu(I(\mathbf{x};\boldsymbol{\omega}))$ and differentiation can be moved inside the integral since Ω is constant. The gradient of the image of rotated events is

$$\frac{\partial I(\mathbf{x};\boldsymbol{\omega})}{\partial\boldsymbol{\omega}} \overset{(4)}{=} -\sum_k \pm_k \nabla\delta\big(\mathbf{x}-\mathbf{x}'_k(\boldsymbol{\omega})\big)\frac{\partial\mathbf{x}'_k}{\partial\boldsymbol{\omega}}, \tag{9}$$

where the derivative of each rotated event (Section III-C) is

$$\frac{\partial\mathbf{x}'_k}{\partial\boldsymbol{\omega}} = \frac{\partial}{\partial\boldsymbol{\omega}}\mathbf{W}(\mathbf{x}_k;\boldsymbol{\omega},t_k-t_0).$$

The gradient of the Dirac delta $\delta(\mathbf{x}) \doteq \delta(x)\delta(y)$ is computed component-wise, $\nabla\delta(\mathbf{x}) = (\delta'(x)\delta(y), \delta(x)\delta'(y))^\top$. We apply finite-differences with step $h = 1$ pixel to approximate the derivative: $\delta'(x) \approx (\delta(x+h/2)-\delta(x-h/2))/h$. Then, the first component of $\nabla\delta$ is approximated by the difference of two 2D deltas like those in (4): $\delta'(x)\delta(y) \approx (\delta(\mathbf{x}-\mathbf{x}_-)-\delta(\mathbf{x}-\mathbf{x}_+))/h$, with $\mathbf{x}_\pm = (\pm h/2, 0)^\top$. A similar argument applies to the second component of $\nabla\delta$. Finally, each 2D delta is implemented by bilinear voting over four pixels, as explained in Section IV.

Also, observe that, in (8), by linearity of the integral (6) and the derivative, both operators commute:

$$\frac{\partial}{\partial\boldsymbol{\omega}}\mu\big(I(\mathbf{x};\boldsymbol{\omega})\big) = \mu\left(\frac{\partial I(\mathbf{x};\boldsymbol{\omega})}{\partial\boldsymbol{\omega}}\right),$$

so the left hand side can be computed once the derivative image (9) has been obtained.

Finally, for the numerically-better behaved contrast that includes Gaussian smoothing (Section IV), the objective function is $\mathrm{Var}\,(I_\sigma(\mathbf{x};\boldsymbol{\omega}))$, with $I_\sigma(\mathbf{x};\boldsymbol{\omega}) = I(\mathbf{x};\boldsymbol{\omega}) * G_\sigma(\mathbf{x})$. The objective gradient has the same form as (8), but with I and $\frac{\partial}{\partial\boldsymbol{\omega}}I(\mathbf{x};\boldsymbol{\omega})$ replaced by I_σ and $\frac{\partial}{\partial\boldsymbol{\omega}}I_\sigma(\mathbf{x};\boldsymbol{\omega}) = \big(\frac{\partial}{\partial\boldsymbol{\omega}}I(\mathbf{x};\boldsymbol{\omega})\big) * G_\sigma(\mathbf{x})$, respectively.

References

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008. 1

[2] T. Delbruck, V. Villeneuva, and L. Longinotti, "Integration of Dynamic Vision Sensor with Inertial Measurement Unit for Electronically Stabilized Event-Based Vision," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2014, pp. 2636–2639. 1, 5

[3] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison, "Simultaneous Mosaicing and Tracking with an Event Camera," in *British Machine Vision Conf. (BMVC)*, 2014, pp. 1–12. 1, 2

[4] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, "Interacting Maps for Fast Visual Interpretation," in *Int. Joint Conf. Neural Networks (IJCNN)*, 2011, pp. 770–776. 1, 2

[5] J. Conradt, "On-Board Real-Time Optic-Flow for Miniature Event-Based Vision Sensors," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, Dec 2015. 2

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, second Edition. 3

[7] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2004. 3

[8] E. Peli, "Contrast in Complex Images," *J. Opt. Soc. Amer. A*, vol. 7, no. 10, pp. 2032–2040, 1990. 4

[9] A. A. Michelson, *Studies in Optics*. New York, NY, USA: Dover, 1995. 4

[10] R. Fletcher and C. M. Reeves, "Function Minimization by Conjugate Gradients," *Comput. J.*, vol. 7, pp. 149–154, 1964. 4

[11] G. Wolberg, *Digital Image Warping*. Washington, DC, USA: Wiley-IEEE Comp. Soc., 1990. 4

[12] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 130dB 3us Latency Global Shutter Spatiotemporal Vision Sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014. 5

[13] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM," *Int. J. Robotics Research*, (In press) 2017. 5

[14] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014, pp. 15–22. 7

[15] J. Engel, J. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Computer Vision (ECCV)*, 2014, pp. 834–849. 7

[16] G. Gallego, A. Yezzi, F. Fedele, and A. Benetazzo, "Variational Stereo Imaging of Oceanic Waves With Statistical Constraints," *IEEE Trans. Image Processing*, vol. 22, no. 11, pp. 4211–4223, Nov. 2013. 8